Test Plan: Brainstorming

**Acceptance Test**

The purpose of this test is to validate and verify that our system meets the design requirements. For instance, for one of our design requirements is that our device displays the heading in degrees within 5° of actual. To test for this we will need to have another compass to compare the output to. For the acceptance test, we will be referencing the engineering requirements, not the marketing requirements.

|  |  |
| --- | --- |
| **Engineering Requirements** | **Justification** |
| 1. The system should be accurate to within 5° of true orientation. | To be a viable compass for use it should have an acceptable accuracy that would not result in an individual or robot being lost at long distances. |
| 2. The system should have a switch to power on the device and within a short time be ready for use. | To save battery life, the user should be able to turn the device on and off. The re-initialization of the device should require as little time as possible so that the user can get their heading and proceed with very little lost time. |
| 3. The LCD display will show a heading (i.e. "N", "SW") and degrees from North between 0 and 360 degrees. | Without an actual moving compass face, the user will need to easily understand the output of the device. Having eight segments it will be easy for the user to visualize their bearing. |
| 4. Should be prepared for later versioning in firmware that will give additional options and restore to factory default through a USB interface. | If an unforeseen software glitch were to arise the device will still possibly be useful if restored to factory settings. The device should be capable of a software update to correct any unforeseen glitches. |
| 5. LCD screen will have a back light so that the compass can be used in light or dark. | Needs to be used in varying conditions. |

1. The group will use our compass and compare the output with a smartphone compass app to ensure that our design will stay within a 5° tolerance as set by the project requirements
2. To test for this requirement, the group has implemented a switch into the design that disconnects power to the system. To test for the amount of time it takes to initialize the device, a stopwatch (on a smartphone) will be used. A finite initialization was not defined in the requirements.
3. This requirement is self-explanatory, the LCD displays the correct information or it does not. This can again be tested with the use of a smartphone compass app to verify that the LCD is displaying the correct heading in degrees as well as the arrow pointing in the right direction.
4. To verify the 4th requirement, we can have two sets of coding. One will be missing some functionality and the other will include it. To verify the requirement, we will implement the change via USB.
5. The group may be getting rid of the 5th requirement. But if we don’t, we can simply toggle the LCD backlight in a dark room to confirm visibility under dark conditions.

**Integration Test**

The purpose of an integration test is to test the sub-modules, which have been individually tested previously, to see how they interact with each other. This is where we verify that the communication between modules is correct. For example, we should test the output of the PMIC under three conditions: battery only, USB only, and both. With the implemented test points we can measure the output of the PMIC with a multi-meter. We still need to consider the engineering requirements when developing the integration tests.

PMIC – The Power Management Integrated Circuit (PMIC) outputs a constant and stable 3.3V to the system. The group will need to verify that all modules receive power from the PMIC. This is different than testing the output of the PMIC as it is loaded while under operation. The group needs to ensure that all modules get power.

Microcontroller – Testing for this should be ongoing while writing the code. Ben or Michael will need to help me write this up.

Magnetometer/Accelerometer – Testing for this should be ongoing while writing the code. Ben or Michael will need to help me write this up.

LCD – Testing for this should be ongoing while writing the code. Ben or Michael will need to help me write this up.

**Unit Test**

The purpose of a unit test is to individually test a sub-module for its desired behavior. This is an isolated test of the sub-module and should utilize ‘stubs’ to model the behavior of the other sub-modules the module-under-test interfaces with. For example, to do a unit test on the magnetometer, the group can use the microcontroller as a stub to output a blinking LED, where the number of LED flashes is 1/10 of the degree heading.

PMIC – Test the PMIC output voltage under three conditions: with battery, with USB, and with both the battery and USB. The charging functionality will also have to be tested. The group will have to drain a battery, check the voltage with a multi-meter, then charge it, and check the voltage after some amount of time charging, say two hours.

Microcontroller – blink some LED’s! Ben or Michael will need to help me write this up.

Magnetometer/Accelerometer – Using the microcontroller as a stub, take in heading values from the magnetometer and make an LED blink for degrees/10. For example, if the heading if the heading is 10° the LED will blink once. If the heading is 160°, the LED will blink 16 times.

LCD – Using the microcontroller as a stub, make the display output desired images or text. Something like a “hello world.” Once this initial step is accomplished, some of the bitmaps from our compass library can be loaded onto the LCD to make sure that they look right.

**Debugging**

The debugging process is ongoing and constant during the development of the system. Since I’m not doing any of the coding, I will need the help of Ben and Michael for this section of the test plan.